

version of "Mortal Combat" by virtue of an association between its module name in table T with an address in data stream 305.

As another example, suppose that terminal 315 needs to download the video game "Asteroids". Of the three versions of "Asteroids" included in table T, the first version cannot be downloaded by terminal 315 because it is only compatible with terminals manufactured by manufacturer A. However, either of the second two versions would be compatible with the requirements included in table 315a, the only difference being that the second version requires only 500 kilobytes to be downloaded, while the third version requires 1 megabyte. In accordance with various aspects of the invention, terminal mechanism 315b may select the "best" version, which would presumably correspond to the 1 megabyte version (i.e., it is the version having the highest degree of match).

As yet another example, note that any of terminals 313, 314 and 315 may download the data table (last entry in table T), but that terminal 316, which executes an operating system manufactured by manufacturer Z, is not compatible with the data table. Thus, it is apparent that each terminal can select for itself—without communicating to downloading source 302—which of several versions of a software or data module can be downloaded into that terminal.

FIG. 4 shows steps which may be executed by each terminal in order to automatically upgrade the operating system to the latest version upon boot-up or other initialization sequence. Beginning in step 401, the terminal boots up or otherwise commences an initialization sequence. This could occur upon power-up, for example, or upon receiving a command from the network.

In step 402, the terminal receives the module descriptor table from the network from a known channel and location. In step 403, the terminal compares all the operating system modules in the received descriptor table which are compatible with the particular hardware configuration of the terminal (e.g., it matches terminal manufacturer, model, operating system manufacturer, and memory). Based on the comparison between the operating system modules in the received table and the hardware parameters in its internal configuration table, the terminal finds the highest operating system version number which is compatible with the terminal's hardware parameters.

In step 404, a comparison is made between the highest compatible operating system version number available in the received descriptor table and the currently executing operating system version number in the terminal. If the highest available operating system version is less than or equal to the version currently executing in the terminal, then in step 405 the terminal continues its normal processing. However, if a newer (higher) version of the operating system is available, in step 406 the terminal downloads the newer version from the download data stream, and reboots the terminal in step 407 to install the newer version. The downloading in step 406 can be accomplished by tuning to a "carousel" channel and locating the newest operating system version according to its address in the data channel (the module address can be extracted from the descriptor table).

In step 404, instead of determining whether a newer version of the operating system is available, the terminal could instead make a determination that the currently loaded operating system is corrupted or otherwise not suitable. In such a circumstance, the terminal could automatically obtain the latest version of the operating system by locating and downloading it from the data stream. Instructions to execute the steps shown in FIG. 4 could be installed into ROM to prevent erasure or corruption.

FIG. 5 shows how a terminal may download an application module or data module in response to selecting an application, whether by user input or other means (e.g., the operating system may select a program or data table for a purpose). Beginning in step 501, an application module (such as a video game, a stock market tool, a billing program, or the like) or a data module (such as a tuning table, operating system parameter table, screen icon or the like) is selected. In step 502, the terminal extracts the module descriptor table from the network at a known channel and location. In step 503, the terminal finds the "best" version of the selected application or data module in the received descriptor table which is compatible with the terminal's capabilities and interfaces. This may include checking all of the fields shown in FIG. 3, a subset of these fields, or additional fields.

In step 504, a comparison is (optionally) made between the "best" located version of the selected module and any currently loaded version (if any) of the module in the terminal. If a currently loaded version of the module is already the "best" version for the terminal, then in step 505 the currently loaded version is executed. However, if a newer version (or a version not currently loaded) is available, then in step 506 the specified version of the module is downloaded from the network. Thereafter, in step 507, the application is executed or the downloaded data module is accessed.

In step 501, instead of an application or data module being "selected", the terminal may perform an equivalent step by terminating an existing application. Thus, for example, if a "navigator" program is normally executed by the terminal to allow a user to select from among a group of applications, then at the completion of any particular application, the terminal could automatically execute the steps shown in FIG. 5 to determine if a newer version of the navigator program were available. Such a case might occur where new features were added to a main menu, for example. Additionally, although not explicitly shown in FIG. 5, the terminal could free up memory areas if needed before downloading newer modules.

It is apparent that many modifications and variations of the present invention are possible, and references to specific values are by example only. As one example, the module descriptors and download data stream could be transmitted periodically, continuously, or a periodically, using either in-band techniques, out-of-band techniques, or over a combination of channels. Either the module descriptor table, the downloaded modules, or both, could be compressed, encoded, encrypted, or otherwise manipulated prior to transmission. Moreover, different modules to be downloaded could be transmitted over different channels, and the descriptor table could include an identifier indicating where the downloadable module could be found. An authentication function can be provided in each terminal to ensure that only authenticated modules are downloaded into the terminal. Finally, although the invention has application to cable television networks, the term "network" is intended to include satellite transmission networks, radio transmission means, and other communication media. It is, therefore, to be understood that within the scope of the appended claims the invention may be practiced otherwise than as specifically described.

What is claimed is:

1. A system for downloading information, comprising: a downloading source comprising a plurality of different modules each corresponding to a different set of terminal compatibility requirements which must be satis-